Mitigating bias in employment offer prediction

Module Name: COMP2261
Date: 18/04/2022
Submitted as part of the degree of BSc Natural Sciences to the
Board of Examiners in the Department of Computer Sciences, Durham University

1 Introduction

MANY companies currently use classification models to determine whether or not a job should be offered to an applicant. While this eases the decision making process on employers, it can lead to certain groups of people not being given a fair chance. In this report, we will explore which groups can often be discriminated against in such models, and attempt to remove this through adversarial bias mitigation.

Throughout our investigation, we will use a recruitment data set from an anonymous company. For each of the 280 individuals in the data set, the following information is stored:

- Applicant code (a unique number for each applicant)
- Gender (1 = male, 2 = female)
- Black, Asian, or Minority Ethnic (BAME) (1 = yes, 2 = no)
- Shortlisted for interview (0 = rejected, 1 = shortlisted)
- Interviewed (0 = not interviewed, 1 = interviewed)
- Female on interview panel (1 = no, 2 = yes)
- Offered (0 = not made an offer, 1 = made an offer)
- Accepted (0 = declined offer, 1 = accepted offer)
- Joined (0 = joined company, 2 = did not join company).

2 Investigating bias in the data set

Before we can attempt to create a fair classification model, we first need to determine which of the aforementioned features are sensitive, with respect to the predictor variable "Offered", indicating whether or not an individual was given a job offer.

2.1 Identifying the sensitive attribute

The two possible sensitive attributes from this data set are "Gender" and "BAME". We can mathematically conclude which of them are sensitive using a χ^2 independence test, where we define the hypotheses as follows:

- Null hypothesis, H_0: attribute and "Offered" variable are independent
- Alternative hypothesis, H_1: there is a relationship between attribute and "Offered" variable,

testing at a significance level of 5%. Table 1 shows the observed and expected values (under H_0) for each gender-offer combination in the data. From this table, we use the chisquare() function from Scipy.stats to obtain a test statistic of 20.54, and a p-value of 5.8×10^{-6} on 1 degree of freedom. We therefore reject the null hypothesis, concluding that "Gender" and "Offered" are not independent variables, meaning "Gender" is a sensitive attribute. It is not difficult to see which group is privileged in this case, as the observed number of males who were offered a job far exceeds its expected value. Similarly, the observed number of females who were offered a job is less than half the expected value. Hence, males are the privileged group.

When we do the same procedure for the "BAME" variable, we find the p-value to be 0.099, hence we do not reject the null hypothesis in this case, meaning there is not sufficient evidence to say that there are privileged and unprivileged groups for this variable.

TABLE 1
Observed and (expected) values for each gender-offer combination from the data set

	Offered	Not offered
Male	18 (7.8)	60 (70.2)
Female	10 (20.2)	192 (181.8)

2.2 Exploring the disparity between males and females

We have discovered that "Gender" is a sensitive attribute in our data set, and determined that it is males who are privileged. To explore this disparity, we can analyze some of the variables for each gender from our data. The data however is purely categorical, so we are limited in what statistics we can produce.

TABLE 2
Percentages for different variables from the data set in each gender sub-population

	Male	Female
% shortlisted	48.7	24.8
% interviewed	34.6	13.9
% offered	23.1	5.0

As we see in Table 2, there are vast differences in the percentage of shortlisted, interviewed, and offered candidates between males and females, with males being successful far more often in each scenario.

2.3 Proving further bias in the data set

We earlier proved that the data is biased towards males when it comes to whether or not the company makes an offer, however given what we have seen in Table 2, we could investigate bias with respect to other variables, such as the shortlisting process. We define the same hypotheses as before at the 5% level, using "Gender" and "Shortlisted" as the two variables. Table 3 shows the observed and expected values for each combination.

TABLE 3
Observed and (expected, under the null hypothesis) values for each gender-shortlisted combination from the data set

	Shortlisted for interview	Not shortlisted
Male	38 (24.5)	40 (53.5)
Female	50 (63.5)	152 (138.5)

From the above table, we again use Scipy.stats chisquare() function, and find the p-value to be 1.01×10^{-4}, so we therefore reject the null hypothesis and find further bias in the shortlisting process (with males privileged, for the same reasons as before).

We have statistically proven that the data set is biased towards males for several key attributes. In order to somewhat eradicate this in a classification model, we will implement a specific notion of fairness to create an unbiased predictor.

3 Designing a fair classification model

Before we enforce a specific fairness definition on a classifier model, we first create a 'regular' classifier, without any implementation of fairness.

3.1 Training a regular classifier

We use the variables "Gender", "BAME", "Shortlisted", "Interviewed", and "Female on interview panel" as the features, and use the variable "Offered" as the target. We then perform a 70/30 train-test split on the data set, and train a logistic regression model that we designed, based on the solution by Jason Brownlee [1]. Note that we chose to create our own logistic regression model instead of using built-in methods from Scikit-learn, in order to be able to attempt to train a fairer model later on.

3.2 Results on test set

To investigate if the fitted model contains any bias, we split the test set into two subsets: one containing only male individuals, and the other only female. Table 4 shows several metrics obtained from implementing the trained classifier on the male and female test sets, as well as the overall test set.

TABLE 4
Metrics on male, female, and overall test sets from regular logistic regression classifier

	Male	Female	Overall
Size	20	64	84
Accuracy	0.85	0.91	0.89
Precision	0.75	1.0	0.80
Recall	0.60	0.14	0.33
F_1 score	0.67	0.25	0.47
% predicted 1 (offered a job)	20.0	1.6	6.0

We see that the model is very accurate in predicting unseen data overall, and actually performs better on the female test set. Despite this high accuracy of 0.91 on the female test set, it is evident there is bias in the data set. The precision of 0.75 on the male test set, which is significantly lower than that of 1.0 for the female test set, indicates that the model is predicting a large number of false positives for males. Essentially, the model is predicting that a male should be offered the job, when in reality they shouldn't, too often. It is also noteworthy that the model predicted 20% of individuals in the male test set to be offered a job, while only predicting 1.6% of females for the same thing. It is clear that we need to design a fairer model to avoid discrimination.

3.3 Choosing a specific fairness definition to implement

Before we attempt to create a fairer model for predicting job offers, we need to decide which fairness notion we will try to implement into our model. We have several to choose from: demographic parity, equality of odds, and equality of opportunity. In this situation, implementing demographic parity seems like the best choice.

We want the event of being offered a job to be independent of gender, which mathematically corresponds to the probability of a male being offered a job being equal to the probability of a female being offered a job. This is the definition of demographic parity. The best way to measure this from the test data is to observe the proportion of individuals who were offered a job in each sub-population, shown in the last row of Table 4.

To implement this chosen fairness notion, we will use pre-processing and in-processing methods that satisfy the exact definition of demographic parity.

3.4 Pre-processing methods - re-weighing

In order to aim for demographic parity in our new model, we re-weigh each of the samples in the training set. For an individual i, the reweighing process takes into account the observed proportion of individuals for which the gender and offer variables are identical to individual i (called $p^{(i)}_{obs}$), and divides it by the expected proportion of such individuals if gender and offer were independent ($p^{(i)}_{exp}$). Therefore,

individual i is given weight $w^{(i)}$ in the learning process, where

$$w^{(i)} = \frac{p^{(i)}_{obs}}{p^{(i)}_{exp}}. \quad (1)$$

This re-weighing technique aims to achieve demographic parity. With our pre-processing complete, we move on to in-processing methods to enforce demographic parity, which involve implementing an adversarial learning component while training the model.

3.5 In-processing methods - using an adversary

There are several ways to implement fairness notions in AI models while training the data. The one we use is an adversarial method outlined by Zhang, Lemoine, and Mitchell in their paper "Mitigating Unwanted Biases with Adversarial Learning" [2].

While training our target model, we also train an adversary model, which tries to predict the sensitive attribute (gender) based on the predicted class (what the target model tries to predict). However, we do not actually want the adversary to perform well in its predictions, as this would indicate some sort of relationship between the sensitive attribute and the predicted class.

Instead, we update the coefficients of the target model in a way which minimizes the mean squared error (MSE) of the target model, but also does not help the adversary in its predictions. Mathematically, we denote the coefficients of the target and adversary as W and U respectively, and the losses L_P and L_A. Then, at each time step t in the gradient descent, we update U to minimize L_A according to the gradient $\nabla_U L_A$. We then alter W by the following equation:

$$W = W - \frac{R}{N}(\nabla_W L_P - \text{proj}_{\nabla_W L_A} \nabla_W L_P - \alpha \nabla_W L_A), \quad (2)$$

where R is the learning rate, N is the number of individuals in the training set, and α is a varying hyperparameter which we set to $\sqrt{1/t}$. In a regular logistic regression classifier, W would be updated by the equation

$$W = W - \frac{R}{N}\nabla_W L_P, \quad (3)$$

so there is a clear difference in how the new model is trained.

We implemented all of the above by defining and creating a child class object of the initial logistic regression object, whose .fit() function was different to that of its parent class in order to execute the new training method described above.

3.6 Results

Table 5 displays the results of our new model on the same test sets as before, with respect to the same metrics. We see that none of the metrics for the male test set have changed, however the metrics on the female test set indicate some measure of debiasing. The precision has decreased, which suggests more females are being given the class label 1 (offered a job), and we also see a substantial increase in the percentage of females who were given the class label 1 (from 1.6% to 10.9%).

While the percentage of males being given class label 1 is still noticeably larger than that for females, our new model does appear to have reduced the bias towards males somewhat. The notion of demographic parity is not completely satisfied by the model, but given that the size of our train and test sets were relatively small, the results are promising.

TABLE 5
Metrics on male, female, and overall test sets from fair implementation of logistic regression classifier enforcing demographic parity

	Male	Female	Overall
Size	20	64	84
Accuracy	0.85	0.94	0.92
Precision	0.75	0.71	0.70
Recall	0.60	0.71	0.67
F_1 score	0.67	0.71	0.70
% predicted 1 (offered a job)	20.0	10.9	13.1

4 Conclusion

In summary, we found that gender was a sensitive attribute in the recruitment dataset we were given, and that males were clearly the privileged group, not just in terms of receiving an offer, but also across other variables such as being shortlisted.

We then trained a regular logistic regression classifier with no fairness notions enforced, and unsurprisingly found that the results heavily favored males. Upon seeing this, we implemented re-weighing and adversarial debiasing techniques in order to try and enforce demographic parity on our model, resulting in a fairer but still biased model.

A useful metric we can use to display this is the adverse impact ratio (AIR), which consists of dividing the proportion of individuals offered a job in the privileged group (males), by the same proportion in the unprivileged group (females) [3]. By doing the calculations, we find that

$$\text{AIR}_{\text{before}} = 0.078$$
$$\text{AIR}_{\text{after}} = 0.55.$$

So there is a significant increase in the AIR, however it is still not very close to the legally required value of 0.8.

References

[1] "How To Implement Logistic Regression From Scratch in Python" - Jason Brownlee PhD.
https://machinelearningmastery.com/implement-logistic-regression-stochastic-gradient-descent-scratch-python/
[2] "Mitigating Unwanted Biases with Adversarial Learning" - Zhang, Lemoine, Mitchell, 2018.
https://arxiv.org/pdf/1801.07593.pdf
[3] "How to fight discrimination in AI" - Andrew Burt.
https://hbr.org/2020/08/how-to-fight-discrimination-in-ai